Profit based unit commitment for GENCOs using Parallel PSO in a distributed cluster

C.Christopher Columbus* and Sishaj P Simon
National Institute of Technology/Electrical and Electronics Engineering, Tiruchirapalli, Tamil Nadu, India
Email: christoccc@gmail.com, sishajpsimon@nitt.edu

***Abstract*— In the deregulated electricity market, each generating company has to maximize its own profit by committing suitable generation schedule termed as profit based unit commitment (PBUC). This article proposes a Parallel Particle Swarm Optimization (PPSO) solution to the PBUC problem. This method has better convergence characteristics in obtaining optimum solution. The proposed approach uses a cluster of computers performing parallel operations in a distributed environment for obtaining the PBUC solution. The time complexity and the solution quality with respect to the number of processors in the cluster are thoroughly tested. The method has been applied to 10 unit system and the results show that the proposed PPSO in a distributed cluster constantly outperforms the other methods which are available in the literature.**

***Index Terms*—deregulated market, profit based unit commitment, particle swarm optimization, distributed environment, parallel processing, parallel particle swarm optimization.**

I. Introduction

The GENCOs objective is to maximize the profit and to place proper bid in the market. In order to do this generation companies need to determine the schedule and operating points based on the load and price forecasted. The traditional unit commitment problem objective is minimizing the cost of operation subject to fulfillment of demand. However in a deregulated environment the traditional unit commitment objective needs to be changed to maximize the profit with relaxation of the demand fulfillment constraint. This unit commitment is referred to as profit based unit commitment. [1].

A competitive and deregulated framework is replacing traditional and centralized regulation in many electric power systems around the world. With the promotion of deregulation of electric power systems, operation, planning and control aspects in traditional power system need to be changed [2-3]. In this new paradigm, the signal that would enforce a unit's on/off status would be the price, including the fuel purchase price, energy sale price, ancillary service sale price, and so on. There are many solution techniques such as integer programming; dynamic programming, Lagrangian relaxation and genetic algorithms are available to solve the PBUC problem [4-6]. Researchers also presented a review on deterministic, meta-heuristic and hybrid approaches of generation scheduling in both regulated and deregulated power markets [7].

*Corresponding author

The basic idea of LR is to relax the UCP constraints into a small sub-problem, which is much easier to solve, and then coordinated by a master problem via properly adjusting a factor called Lagrangian multiplier. For all that, it has proven to be a very difficult task that may come sometime from oscillation of their solution by slight change of the multiplier. In order to overcome these complex mathematical problems, there are other method of computational methodology, which is shared by popular artificial intelligence such as genetic algorithm and evolutionary programming.

Charles W. Richter et.al presented a PBUC problem formulation using genetic algorithm (GA) which considers the softer demand constraints and allocates fixed and transitional costs to the scheduled hours [6]. Pathom Attaviriyanupap et.al proposed a method that helps GENCO to make a decision on how much power and reserve that should be sold in markets, and how to schedule generators in order to receive the maximum profit [7]. Here the authors have considered both power and reserve generation at the same time. In [8], H.Y. Yamin et.al proposed an auxiliary hybrid model using LR and GA to solve UCP. GA is used to update the Lagrangian multiplier also presented their view on the profit based unit commitment in day- ahead electricity markets considering the reserve uncertainty [9].

The optimization method known as particle swarm optimization (PSO) algorithm developed by Eberhart and Kennedy is successfully applied to solve nonlinear optimization problems. Therefore an attempt is made to solve the PBUC problem using this algorithm. The swarm-based algorithm described in this paper is a search algorithm capable of locating optimal solutions efficiently.

The proposed method is applied to solve PBUC problem with ten-unit and hundred-unit-test systems. The performance of the PPSO algorithm in terms of solution quality is compared with that of other algorithms reported in literature for the above mentioned problem in power system operation. Likewise, simulation results demonstrate the feasibility and effectiveness of the proposed method, as compared with the results available in the literature.

II. Problem formulation

The objective of the PBUC problem is to obtain the optimal unit commitment schedule thereby maximizing GENCOs profit. The problem formulation is given as follows:

$$Maximize \quad PF = RV - TC \tag{1}$$

or

DOI: 01.IJEPE.02.03. 512

$$Minimize \quad TC - RV \tag{2}$$

where PF is the total profit ($), RV is the total revenue ($) and TC is the total cost ($).

Here,

$$TC = \sum_{t=1}^{T}\sum_{i=1}^{N}[C_i(P_{(i,t)}I_{(i,t)}) + ST_t] \tag{3}$$

$$RV = \sum_{t=1}^{T}\sum_{i=1}^{N}[(\sigma_g(t)P_{(i,t)}I_{(i,t)})] \tag{4}$$

where C_i is the production cost which is calculated by using the equation (10). $P_{(i,t)}$ is the power level of i^{th} generator unit at t^{th} hour (MW), $I_{(i,t)}$ is the commitment state of i^{th} unit at t^{th} hour, ST_t is the startup cost ($), t is the index for time, T is the dispatch period in hours, i is the index for generator unit, N is the total number of generating units and $\sigma_g(t)$ is the forecasted market price for energy at time t.

$$C_i(P_{(i,t)}) = a + b*P_{(i,t)} + c*P_{(i,t)}^2 \tag{5}$$

where a, b and c are the fuel cost co-efficients.

System Constraints

Demand constraints

$$\sum_{i=1}^{N}P_{(i,t)}I_{(i,t)} \le D_t \qquad t = 1.....T \tag{6}$$

where D_t is the total system demand at time t.

Here, demand and reserve constraints are different from traditional UC problem because GENCO can now select to produce demand and reserve less than forecasted level if it creates more profit.

Unit constraints

1. Unit power limit

$$P_{i,\min} \le P_{(i,t)}I_{(i,t)} \le P_{i,\max} \tag{7}$$

where $P_{i,\min}$ is the minimum power output of i^{th} generator unit (MW) and $P_{i,\max}$ is the maximum power output of generator unit (MW).

2. Minimum Up and Down time constraints

$$[X^{on}(i,t-1) - T^{on}(i)] * [I_{(i,t-1)} - I_{(i,t)}] \ge 0 \tag{8}$$

$$[X^{off}(i,t-1) - T^{off}(i)] * [I_{(i,t)} - I_{(i,t-1)}] \ge 0 \tag{9}$$

where $X^{on}(i,t)$ is the "On" duration of i^{th} generator unit till time t, $X^{off}(i,t)$ is the "Off" duration of i^{th} generator unit till time t, $T^{on}(i)$ is the minimum up-time of i^{th} generator unit and $T^{off}(i)$ is the minimum down-time of i^{th} generator unit.

III. Parllel clustering environment

In clusters, powerful low cost workstations and/or PCs are linked through fast communication interfaces to achieve high performance parallel computing. Workstation clusters have become an increasingly popular alternative to traditional parallel supercomputers for many workloads requiring high performance computing. The use of parallel computing for scientific simulations has increased tremendously in the last ten years, and parallel implementations of scientific simulation codes are now in widespread use [10, 11]. There are two dominant parallel hardware/software architectures in use today are Distributed memory, and Shared memory. In shared memory systems, parallel processing occurs through the use of shared data structures, or through emulation of message passing semantics in software. Distributed memory systems are composed of a number of interconnected computational nodes, which do not share memory, but can communicate with each other through a high-performance ether net switch (HPES) as shown in Figure 1. Parallelism is achieved on distributed memory systems with multiple copies of the parallel program running on different nodes, sending messages to each other to coordinate computations. The cluster should perform as a parallel computing resource, achieving higher performance than possible using workstations configured in a more standard way. The nodes in the cluster are always used in groups, not individually as in a general purpose workstation laboratory.

Speedup factor and efficiency:

To evaluate the parallel performance of the PPSO algorithm, the speedup factor SW_h and efficiency EW_h of the cluster [12-13] is calculated as follows;

$$SW_h = W_t / W_{ht} \tag{10}$$

$$EW_h = SW_h / W_h \tag{11}$$

where W_t and W_{ht} are the execution time of single processor and cluster respectively.

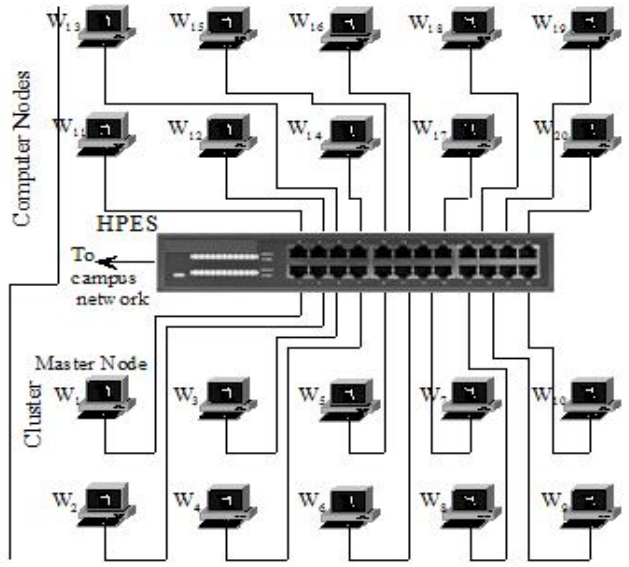

Figure 1. Distributed cluster of workstations (20 Nodes)

DOI: 01.IJEPE.02.03. 512

III. Particle swarm optimization algorithm

Particle Swarm Optimization (PSO) is an optimization technique inspired from bird flocking which is developed by Dr Eberhart and Dr. Kennedy way back in the year 1995 [14]. It is a population based algorithm where each individual (particle) in the population is a potential solution, flies in the D dimensional problem space with a velocity which is dynamically adjusted according to the flying experiences of its own and its colleagues.

A) Standard PSO algorithm

Suppose that the search space is D-dimensional and m particles form the colony. The i^{th} particle represents a D dimensional vector $X_i (i=1, 2\ldots m)$. It means that the i^{th} particle positions at $X_i = (x_{i1}, x_{i2}, \ldots\ldots, x_{iD})$ (i=1, 2… m) in the searching space. The position of each particle is a potential result. The calculation of the particle's fitness is carried out by putting its position value into a designated objective function. When the fitness is higher, the corresponding X_i is "better". The i^{th} particle's "flying" velocity is also a D-dimensional vector, denoted as $V_i = (v_{i1}, v_{i2}, \ldots\ldots, v_{iD})$ Denote the best position of the i^{th} particle as $P_i = (p_{i1}, p_{i2}, \ldots\ldots, p_{iD})$ and the best position of the colony as $P_g = (p_{g1}, p_{g2}, \ldots\ldots, p_{gD})$ respectively. The PSO algorithm can be performed by the following equations (12, 13).

$$V_{id}(k+1) = V_{id}(k) + c_1 r_1 (P_{id}(k) - x_{id}(k)) + c_2 r_2 (P_{gd}(k) - x_{id}(k)) \quad (12)$$

$$x_{id}(k+1) = x_{id}(k) + v_{id}(k+1) \quad (13)$$

Where k represents the iterative number,
c_1, c_2 are learning factors. Usually $c_1 = c_2 = 2$, r_1, r_2 are random numbers between (0, 1).The termination criterion for the iterations are determined according to whether the max generation or a designated value of the fitness of P_g is reached

B) Parallel PSO algorithm

PPSO algorithm is implemented to determine the commitment status of each unit over a scheduled period of (24 hours) time in order to maximize the profit. The procedure of the proposed algorithm to solve PBUCP is as follows.

Step 1: Generator and PSO Parameters Specification

Specify the generator minimum and maximum generation limits, minimum up and down time constraints and start up cost of each unit. Specify the PPSO parameters such as population size (M), inertia weight factor (w), dimension of the system (D), acceleration constants (c_1 and c_2), velocity maximum and minimum limits, maximum iterations (Max iter). Set iteration number iter=1 and time t=1.

Step 2: Particles sharing policy

Master node decides the sharing of particles by particles sharing policy (PSP).Therefore the number of particles allocated in a slave processors or workers is given by

$$PSP = \begin{cases} x, if\ particles\ \ divisible\ by\ W_h \\ x+1\ for\ the\ first\ \ hx\ slaves\ \ and \\ \qquad x\ for\ the\ remaining\ ,\ Otherwise \end{cases} \quad (14)$$

where

$$x = floor(N_{Particles} / W_h) \quad (15)$$

$$hx = N_{Particles} - (x * W_h) \quad (16)$$

Master node allocate (x+1) particles to the first hx slaves in the W_h cluster $(W_1 \ldots W_{hx} \ldots W_h)$ and x ants to the remaining slaves $(W_{hx+1} \ldots W_h)$. Where $N_{Particles}$=population.

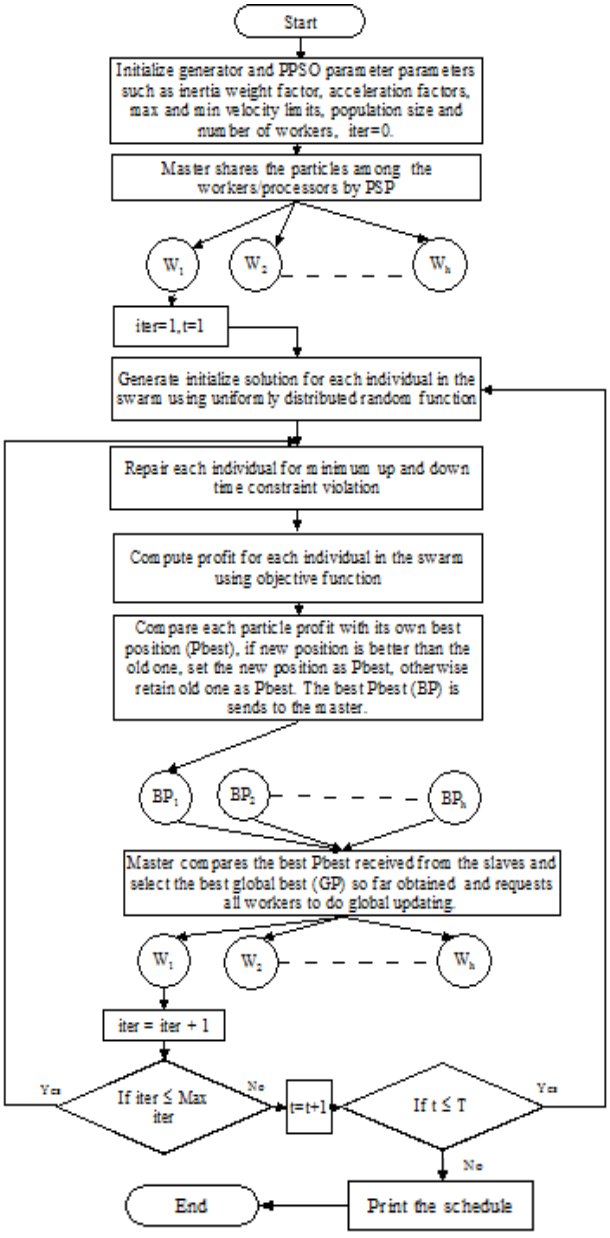

Figure 2. Flowchart of PPSO for PBUC

Step 3: Initialization of Individual in the swarm

The initial solution of each individual $U_j = [u_{n1}\ u_{n2} \ldots u_{nT}]$, (j=1, 2, …,M), (n=1, 2, ….., N)) for complete M population is generated randomly. The position of each unit u_{nt} of each particle is generated using uniformly distributed random

DOI: 01.IJEPE.02.03.512

function, which generates either 0 or 1. Similarly the initial velocity of each particle is generated randomly using uniformly distributed random function, which generates a real value between V_{min} and V_{max}. The representation of each individual for 'N' number of generating units for a scheduled period of time is as follows:

$$U = \begin{bmatrix} u_{11} & u_{12} & . & . & u_{1N} \\ u_{21} & u_{22} & . & . & u_{2N} \\ . & . & . & . & . \\ . & . & . & . & . \\ u_{T1} & u_{T2} & . & . & u_{TN} \end{bmatrix} \quad (17)$$

Step 4: Defining the evaluation function

The merit of each individual particle in the swarm is found using a fitness function called evaluation function. Each particle in the population is evaluated using the objective function given by (1).

Step 5: Repair Minimum up and down time constraints violation

Repair each unit for each particle in the swarm for minimum up and down time constraints violation.

Step 6: Modifying best particle position (Pbest)

To modify the position of each individual in the next stage is obtained from equation (12).

The weighting function is defined as follows

$$w = w_{max} - \left(\frac{w_{max} - w_{min}}{iter_{max}} \right) iter \quad (18)$$

Where,

w_{max} , w_{min} - Initial, final weights

$iter_{max}$ - Maximum iteration number

$iter$ - Current iteration number

To control excessive roaming of particles, velocity is restricted between $-V^{max}$ *and* $+V^{max}$.

The maximum velocity limit for the j^{th} generating unit is computed as follows:

$$V^{max} = \frac{P_j^{max} - P_j^{min}}{R} \quad (19)$$

The particle position vector is updated using equation (13). The values of the evaluation function are calculated for the updated positions of the particles. Evaluate each particle using object function. Compare each particle evolution value with its own best position (Pbest or BS). If the present particle position is better than the old value set new particle position as Pbest, otherwise retain old value.

Step 7: Computation of global best

Master receives the information of local best solutions (BS_1, BS_2,…..,BS_h) from the workers and computes the best solution among them as the global best. Whenever a global best solution is selected by the master, and if the total profit is found to be more than the maximum total profit computed so far, then the present global best is memorized, or else the previous maximum total profit solution is retained as global best. The new global best is sent to the workers and the workers saved the received global best as their global best.

Step 8: Memorize the best solution obtained so far and increment iteration number. Stop the process if iteration number is equal to the maximum number of cycles. Otherwise go to step 4.

Step 9: Increment the time and repeat step 3 to step 8 for the given scheduled period (24 hours) of time.

The flowchart of the proposed method is shown in the Figure 2.

IV. Numerical results

The PABC method for PBUC is first tested on 10 unit system available in the literature as Case 1. It is also validated on multiple test systems of 100 units in Case 2. The parallel computation is carried out in the MATLAB® environment of R2007b using distributing computing toolbox. The parallel computation is carried out through distributed memory environment. In a distributed environment, a cluster with the maximum size of 20 nodes/processors (Pentium - IV 3.40 GHz, 1GB RAM) is used.

A. Case 1: 10 Unit System

In order to participate in the market, GENCOs have to prepare a self commitment according to the forecasted load and price. In this case, the commitment schedule is prepared to maximize the GENCOs' profit by calculating the generator coefficients with the satisfaction of constraints. Here the profit of the company gets the first priority and the demand satisfaction is not mandatory. So, GENCOs will make the self commitment depending upon the forecasted price to get surplus profit. The test system consists of 10 generating units. Here the generating unit data and load data are taken from [15]. The constraints included for PBUC in [15] are considered. Based on the forecasted market price of energy information, the proposed PPSO model is used to generate dispatch schedule for 24 hours time horizon. The dispatch schedule of ten unit system is given in TABLE I. Optimal parameters obtained by trial and error for PPSO is as follows: Population size=200, Acceleration coefficients, c1=0.2, c2=0.2, Inertia weight: W_{max}=0.9, W_{min}=0.2 and Maximum iteration=300.

The comparison of the proposed method with other existing methods given in TABLE II proves that PPSO gives better solution, i.e. a difference in profit of $1260.23 is achieved when compared to Muller method [15]. The time taken to get the best schedule is 168 sec. The PPSO yields a higher profit of 1.2 % than the Muller method. Figure 3 shows the execution time achieved by the different cluster sizes. The execution time will reduce, when the cluster size increases. TABLE III shows the speedup factor and efficiency achieved by different sizes of cluster. When the cluster size increases the speedup factor also increases. i.e., performance of the cluster will increase, when its size increases.

DOI: 01.IJEPE.02.03.512

TABLE I. DISPATCH SCHEDULE FOR 10 UNIT SYSTEM

Hr	Load (MW)	U1 (MW)	U2 (MW)	U3 (MW)	U4 (MW)	U5 (MW)	U6 (MW)	U7 (MW)	U8 (MW)	U9 (MW)	U10 (MW)	Fuel Cost ($)	Transition Cost ($)	Total Cost ($)	Revenue ($)	Profit ($)
1	700	455	245	0	0	0	0	0	0	0	0	13683.13	0	13683.13	15505.00	1821.87
2	750	455	295	0	0	0	0	0	0	0	0	14554.50	0	14554.50	16500.00	1945.50
3	850	455	395	0	0	0	0	0	0	0	0	16301.89	0	16301.89	19635.00	3333.11
4	950	455	455	0	0	0	0	0	0	0	0	17353.30	0	17353.30	20611.50	3258.20
5	1000	455	455	0	0	0	0	0	0	0	0	17353.30	0	17353.30	21157.50	3804.20
6	1100	455	385	130	130	0	0	0	0	0	0	21879.33	2220	24099.33	25245.00	1145.67
7	1150	455	435	130	130	0	0	0	0	0	0	22755.04	0	22755.04	25875.00	3119.96
8	1200	455	455	130	130	0	0	0	0	0	0	23105.76	0	23105.76	25915.50	2809.74
9	1300	455	455	130	130	130	0	0	0	0	0	26184.02	1800	27984.02	29640.00	1655.98
10	1400	455	455	130	130	162	68	0	0	0	0	28768.21	340	29108.21	41090.00	11981.79
11	1450	455	455	130	130	162	80	0	0	0	0	29047.98	0	29047.98	42571.80	13523.82
12	1500	455	455	130	130	162	80	0	0	0	0	29047.98	0	29047.98	44689.80	15641.82
13	1400	455	455	130	130	162	68	0	0	0	0	28768.21	0	28768.21	34440.00	5671.79
14	1300	455	455	130	130	110	20	0	0	0	0	26588.96	0	26588.96	31850.00	5261.04
15	1200	455	455	130	130	0	0	0	0	0	0	23105.76	0	23105.76	26325.00	3219.24
16	1050	455	335	130	130	0	0	0	0	0	0	21005.17	0	21005.17	23415.00	2409.83
17	1000	455	285	130	130	0	0	0	0	0	0	20132.56	0	20132.56	22250.00	2117.44
18	1100	455	385	130	130	0	0	0	0	0	0	21879.33	0	21879.33	24255.00	2375.67
19	1200	455	455	130	130	0	0	0	0	0	0	23105.76	0	23105.76	25974.00	2868.24
20	1400	455	455	130	130	0	0	0	0	0	0	23105.76	0	23105.76	26500.50	3394.74
21	1300	455	455	130	130	0	0	0	0	0	0	23105.76	0	23105.76	27027.00	3921.24
22	1100	455	385	130	130	0	0	0	0	0	0	21879.33	0	21879.33	25245.00	3365.67
23	900	455	445	0	0	0	0	0	0	0	0	17177.91	0	17177.91	20475.00	3297.09
24	800	455	345	0	0	0	0	0	0	0	0	15427.42	0	15427.42	18040.00	2612.58
Total ($)												525316.37	4360	529676.37	634232.60	104556.23

TABLE II. COMPARISON OF PBUC SOLUTIONS (10 UNIT SYSTEM)

Method	Profit ($)
TS-RP[16]	101086.00
TS-IRP[16]	103261.00
Muller method[15]	103296.00
PSO	104356.00
PPSO	104556.23

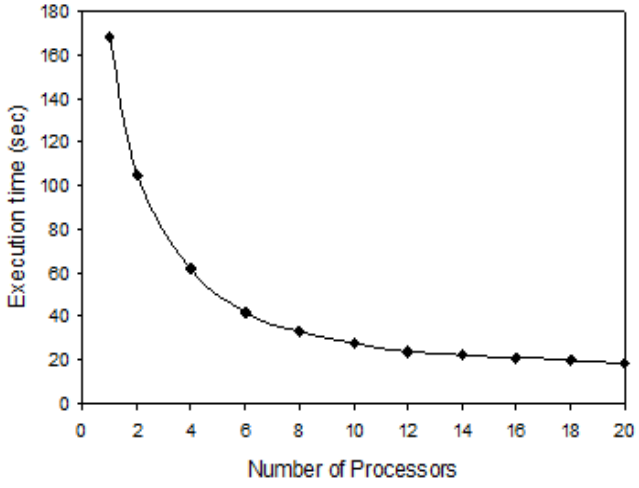

Figure 3. Execution time chart for 10 unit system

TABLE III. COMPARISION OF SPEEDUP FACTOR AND CLUSTER EFFICIENCY (10 UNIT SYSTEM)

W_h	2	4	6	8	10	12	14	16	18	20
Speedup factor	1.6	2.69	4.01	5.04	6.03	7.01	7.49	8.11	8.43	8.98
EW_h(%)	80.01	67.25	66.83	63.01	60.03	58.41	53.51	50.68	46.83	44.91

B. Case 2: 100 Unit Syatem

This test system consists of multiple generating units such as 100 generating units. More number of generating units is considered in order to validate the feasibility of the application of PPSO for large scale power system. The data for different groups of generating units are obtained by duplicating the 10 unit system data. The demand is multiplied with respect to the system size; however the generating limits, the minimum up/down time constraints remain the same. Based on the forecast market price of energy information, the proposed PPSO model is used to generate dispatch schedule for 24 hours time horizon. The parameter setting of the 10 unit system is extended for the multiple test systems.

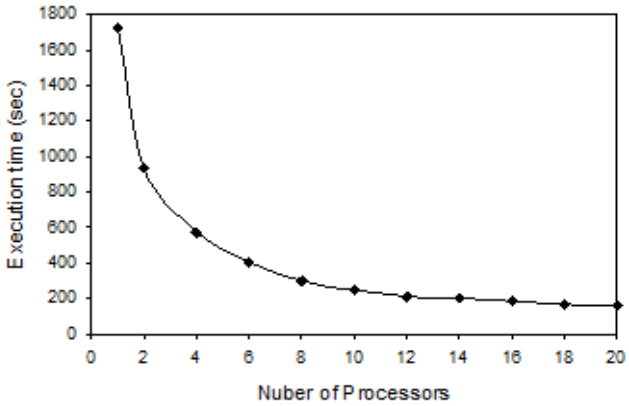

Figure 4. Execution time chart for 100 unit system

DOI: 01.IJEPE.02.03.512

TABLE IV. COMMITMENT STATUS FOR 100 UNIT SYSTEM

Hr	Unit status	Fuel Cost ($)	Transition Cost ($)	Total Cost ($)	Revenue ($)	Profit ($)
1	1100000000110000000011000000001100000000110000000011000000001100000000 1100000000110000000011000000001100000000	136831.30	0	136831.30	155050	18218.70
2	1100000000110000000011000000001100000000110000000011000000001100000000 1100000000110000000011000000001100000000	145545.00	0	145545.00	165000	19455.00
3	1100000000110000000011000000001100000000110000000011000000001100000000 1100000000110000000011000000001100000000	163018.90	0	163018.90	196350	33331.10
4	1100000000110000000011000000001100000000110000000011000000001100000000 1100000000110000000011000000001100000000	173533.00	0	173533.00	206115	32582.00
5	1100000000110000000011000000001100000000110000000011000000001100000000 1100000000110000000011000000001100000000	173533.00	0	173533.00	211575	38042.00
6	1111000000111100000011110000001111000000111100000011110000001111000000 1111000000111100000011110000001111000000	218793.31	22200	240993.30	252450	11456.69
7	1111000000111100000011110000001111000000111100000011110000001111000000 1111000000111100000011110000001111000000	227550.41	0	227550.40	258750	31199.59
8	1111000000111100000011110000001111000000111100000011110000001111000000 1111000000111100000011110000001111000000	231057.59	0	231057.60	259155	28097.41
9	1111100000111110000011111000001111100000111110000011111000001111100000 1111100000111110000011111000001111100000	261840.21	18000	279840.20	296400	16559.79
10	1111110000111111000011111100001111110000111111000011111100001111110000 1111110000111111000011111100001111110000	287682.13	3400	291082.10	410900	119817.90
11	1111110000111111000011111100001111110000111111000011111100001111110000 1111110000111111000011111100001111110000	290479.78	0	290479.80	425718	135238.20
12	1111110000111111000011111100001111110000111111000011111100001111110000 1111110000111111000011111100001111110000	290479.78	0	290479.80	446898	156418.20
13	1111110000111111000011111100001111110000111111000011111100001111110000 1111110000111111000011111100001111110000	287682.13	0	287682.10	344400	56717.87
14	1111110000111111000011111100001111110000111111000011111100001111110000 1111100000111110000011111000001111100000	263368.00	0	263368.00	318500	55132.00
15	1111100000111110000011111000001111100000111110000011111000001111100000 1111100000111110000011111000001111100000	231057.59	0	231057.60	263250	32192.41
16	1111100000111110000011111000001111100000111110000011111000001111100000 1111100000111110000011111000001111100000	210051.71	0	210051.70	234150	24098.29
17	1111100000111110000011111000001111100000111110000011111000001111100000 1111100000111110000011111000001111100000	201325.61	0	201325.60	222500	21174.39
18	1111100000111110000011111000001111100000111110000011111000001111100000 1111100000111110000011111000001111100000	218793.31	0	218793.30	242550	23756.69
19	1111100000111110000011111000001111100000111110000011111000001111100000 1111100000111110000011111000001111100000	231057.59	0	231057.60	259740	28682.41
20	1111100000111110000011111000001111100000111110000011111000001111100000 1111100000111110000011111000001111100000	231057.59	0	231057.60	265005	33947.41
21	1111100000111110000011111000001111100000111110000011111000001111100000 1111100000111110000011111000001111100000	231057.59	0	231057.60	270270	39212.41
22	1111100000111110000011111000001111100000111110000011111000001111100000 1111100000111110000011111000001111100000	218793.31	0	218793.30	252450	33656.69
23	1100000000110000000011000000001100000000110000000011000000001100000000 1100000000110000000011000000001100000000	171779.10	0	171779.10	204750	32970.90
24	1100000000110000000011000000001100000000110000000011000000001100000000 1100000000110000000011000000001100000000	154274.20	0	154274.20	180400	26125.80
Total ($)		5250642.	43600	5294242.	634232	1048083.

TABLE V. COMPARISION OF SPEEDUP FACTOR AND CLUSTER EFFICIENCY (100 UNIT SYSTEM)

W_h	2	4	6	8	10	12	14	16	18	20
Speedup factor	1.85	3.01	4.23	5.71	6.98	8.14	8.62	9.28	10.39	11.05
EW_h(%)	92.51	75.25	70.51	71.37	69.81	67.83	61.57	58.01	57.72	55.25

The Commitment status of 100 unit system is given in TABLE IV. Figure 4 shows the execution time achieved by the different cluster sizes. The execution time will reduce, when the cluster size increases. TABLE V shows the speedup factor and efficiency achieved by different sizes of cluster. When the cluster size increases the speedup factor also increases. i.e., performance of the cluster will increase, when its size increases. When a single processor is used, it consumes more time for execution, i.e., 1728.01 sec for 100 units. The execution time for the 20 node cluster of 100 units are around 156.38 sec. It clearly shows the execution time decreases as the number of processor increases. Each test system has been tested for 30 trial runs and the best results are presented.

DOI: 01.IJEPE.02.03.512

V. Conclusions

This paper proposes a MPI based PPSO model for PBUC, computing in parallel, in a distributed environment. The approach is simple, efficient, and economic and can be extended for making smarter decisions in a large scale power system. Simulation results obtained from the cluster demonstrate the accuracy of the proposed algorithm and its capability of greatly reducing the execution time.

References

[1] Eric H. Allen and Marija D. Ilic, "Reserve markets for power systems reliability", *IEEE Trans. on Power Systems*, vol. 15 no 2, pp. 228-233, 2000.

[2] M. Shahidehpour and M. Mawali, *Maintenance scheduling in restructured power systems*, Nowell, MA Kluwer, 2000.

[3] G.B. Sheble and G.N. Fahd, "Unit commitment literature synopsis", *IEEE Trans. on Power Systems*, vol. 9, pp. 128-135, 1994.

[4] Narayana Prasad Padhy, "Unit commitment - A bibliographical survey", *IEEE Trans. on Power Systems*, vol. 19, no. 2, pp. 1196-1205, 2004.

[5] Narayana Prasad Padhy, "Unit commitment problem under deregulated environment- a review", *Power Engineering Society General Meeting*, 2, pp. 1088-1094, 2003.

[6] Charles W. Richter and Gerald B. Sheble, "A Profit based Unit Commitment GA for Competitive Environment", *IEEE Trans. on Power Systems*, vol. 15, no. 2, pp. 715-721, 2000.

[7] Pathom Attaviriyanupap, Hiroyuki Kita, Eiichi Tanka and Jun Hasegawa, "A hybrid LR-EP for solving new profit –based UC problem under competitive environment", *IEEE Transaction on Power Systems*, vol.18, no. 1, pp. 229-237, 2003.

[8] H.Y. Yamin and S.M. Shahidehpour, "Unit commitment using a hybrid model between Lagrangian relaxation and genetic algorithm in competitive electricity markets", *Electric Power Systems Research*, vol. 68, pp. (83-92, 2004.

[9] I. Jacob Raglend, C. Raghuveer, G. Rakesh Avinash, N.P. Padhy and D.P. Kothari, "Solution to profit based unit commitment problem using particle swarm optimization", *Applied soft computing*, vol. 10, pp. 1247-1256, 2010.

[10] Dingju Zhu and Jianping Fan, "Application of parallel computing in digital city", *The 10th IEEE International Conference on High Performance Computing and Communications*, pp. 845-848, 2008.

[11] Weiwei Lin, Changgeng Guo, Deyu Qi, Yuchong Chen and Zhang Zhili, "Implementations of grid-based distributed parallel computing", *First International Multi-Symposiums on Computer and Computational Sciences*, pp. 312-317, 2006.

[12] H. T. Kumm and R. M. Lea, "Parallel computing efficiency: climbing the learning curve", *TENCON'94*, pp. 728-738, 1994.

[13] X.-H. Sun, L.M. Ni, "Scalable problems and memory-bounded speedup", *Journal of Parallel and Distributed Computing*, vol. 19, no. 1, pp. 27–37, 1993.

[14] J. Kennedy and R. C. Eberhart, "Particle swarm optimization," *IEEE international Conference on Neural Networks*, vol. 4, pp.1942-1948, 1995.

[15] Chandram K, Subrahmanyam N, Sydulu M, "New approach with Muller method for profit based unit commitment", *Power and Energy Society General Meeting - Conversion and Delivery of Electrical Energy in the 21st Century*, pp. 1-8, 2008.

[16] Victoire T. A. A, Jeyakumar A. E, "Unit commitment by a tabu-search-based hybrid-optimization technique", *IEE Proc. Gener. Transm. Distrib.*, vol. 152, pp. 563-570, 2005.

Authors biography

C. Christopher Columbus was born in India and received his Bachelors of Engineering (Electrical and Electronics Engineering) in M. S University, Tirunelveli and Masters of Engineering (Computer Science and Engineering) at Anna University, Chennai, India in the years 1998 and 2005 respectively. He is currently pursuing his research degree in the Department of Electrical and Electronics Engineering, National Institute of Technology, Tiruchirappalli, Tamil Nadu, India. His research interest includes Deregulation of Power system and Parallel computing applications in Power Systems.

Sishaj Pulikottil Simon was born in India and received his Bachelors of Engineering (Electrical and Electronics Engineering) and Masters of Engineering (Applied Electronics) at Bharathiar University, Coimbatore, India in the years 1999 and 2001 respectively. He obtained his Ph.D., (Power System Engineering) at Indian Institute of Technology (IIT), Roorkee, India in 2006. Currently, he is an Assistant professor in the Department of Electrical and Electronics Engineering at National Institute of Technology (NIT), Tiruchirappalli, Tamil Nadu, India.

DOI: 01.IJEPE.02.03.512